Python Programmierung 2

Höhere Fachschule Uster

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
1. Abend	Vertiefen des Klassen-Konzept in Python	• Repetition class(), methoden, instance-variablen, properties • private / public • Statische Methoden • docStrings • Vererbung • Reflaction (Zugriff auf docStrings, dynamisch Function-Call) • Enumerations (ChatJPT: *Good morning. Pls show me some examples for enums in Python*) • Sub-Classes (e.g. Sense_Hat)	Problem-Based	3h 50’	Code-Refactoring

2. Abend	Refactoring Log-ger-Klasse	• Requirements überarbeiten
		• Interface (Abwärtskompatible) überarbeite (Cleancode)
		  ○ Public/private/Properties/setter/getter
		  ○ Naming
		  ○ ENUM für Log-Level
		  ○ docStrings / API Doc
		
		⇨ Refactoring Weather Application (Usage of Logger-Klasse)
			Problem-Based	3h 50’	Code-Refactoring

Python Programmierung 2

Höhere Fachschule Uster

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben
3. Abend	Umgang mit Modulen und Packages Exception Handling in Python	**Leistungskontrolle 1 (Design and Implement a Class)** • Unterschied Packages / Module • Dokumentation von Modulen • Namespaces • Selektiver Import • Abfangen von Exceptions: Try / except / else / finally • Exception werfen: Raise • Exception-Handling: lösen/werfen • Eigene Exceptions	Problem-Based	1h 2h 50’	Code-Refactoring
4. Abend	Multithreadfing in Python	• Threads kreieren (fork) / stoppen • Kommunikation zwischen Threads • Thread-Synchronisation • Locks • I/O Interrupts • Time-Interrupts	Problem-Based	3h 50’	Code-Refactoring

Python Programmierung 2

Höhere Fachschule Uster

Unterrichtsplan

Walter Rothlin

Abend	Lernziel	Thema / Inhalt	Methode	Zeitbedarf	Hausaufgaben

5. Abend	Hardware ansteuern	**Leistungskontrolle 2 (Design and Implement a Sub-Class)**
		• GPIO (LED-Steuerung)
		• PiPlates
		  ○ Relais-Karte
		  ○ Analog/digital I/O
		⇨ Mutteruhr für SBB-Uhr
		⇨ Selecta-Automat
			Problem-Based	1h
				2h 50’
					Code-Refactoring

6. Abend	Template Mechanismus	• Template-Mechanism (Jinja2)
		  ○ ChatJPT: *Give me an example for JINJA Templates?*
		  ○ ChatJPT: *Can I also implement loops and if-then-else in templates?*
			Problem-Based	3h 50’	Code-Refactoring

7. Abend	Web-Anwendung mit FLASK	• Hello-Word with FLASK • Request / Responses • Static HTML-content • Mimetype und JSON responses • Parameter Übergabe (get/put) • Endpoints	Problem-Based	3h 50’	Code-Refactoring
8. Abend	MLZ	**Modullernzielkontrolle (MLZ)**	Ein lauffähige Applikation Nach Vorgaben unter Zeitdruck implementieren.	4h	

9. Abend	Fachgespäche	Einzelne Fachgespäche über MLZ Selbständiges Arbeiten an: • XML processing in Python (XPath, Schemas, XSLT) • Excel read/write access • RegEx		3h 50’	

Bemerkungen:

- Jeder Abend dauert 4 Lektionen.
- Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden.
- Die Studierenden lösen die Übungen auf ihren privaten Notebooks und dem eigenen Raspberry.
- Der Leistungsnachweis am 8.Abend ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen